

HỆ THỐNG THÔNG TIN QUẢN LÝ

Cấu trúc dữ liệu theo mô hình thực thể - liên kết ER

Mô hình thực thể - liên kết (Entity Relationship Model – ER Model)

- ➔ Mô hình dữ liệu thực thể kết hợp (E-R - entity-relationship data model) do Peter Pin-Shan Chen đề xuất năm 1976, nhìn thế giới thực như là một tập các đối tượng căn bản được gọi là các *thực thể*, và *các mối quan hệ* ở giữa các đối tượng này.
- ➔ Mô hình ER được dùng để xây dựng mô hình dữ liệu ý niệm (Conceptual data modeling) nhằm biểu diễn cấu trúc và các ràng buộc của CSDL.
- ➔ Mô hình ER như một công cụ để trao đổi ý tưởng giữa nhà thiết kế và người dùng cuối trong giai đoạn phân tích. Nó độc lập với DBMS và quá trình thi công database.

Mô hình thực thể - liên kết (Entity Relationship Model – ER Model)

⇒ Mục đích của mô hình E – R:

- Làm thống nhất quan điểm về dữ liệu của những người tham gia hệ thống gọi là **quy tắc nghiệp vụ** (business rule) : Người quản lý, người dùng cuối, người thiết kế hệ thống
- Xác định các xử lý về dữ liệu cũng như các ràng buộc (constraint) trên các dữ liệu.
- Giúp đỡ việc thể hiện cơ sở dữ liệu về mặt cấu trúc: Sử dụng thực thể và các mối liên kết giữa các thực thể. Biểu diễn mô hình quan hệ thực thể bằng một sơ đồ.

Quá trình thiết kế mô hình dữ liệu ý niệm

- ⇒ **Bước 1:** Nhận dạng các kiểu thực thể
- ⇒ **Bước 2:** Nhận dạng các kiểu liên kết giữa các thực thể
- ⇒ **Bước 3:** Nhận dạng các thuộc tính của các kiểu thực thể và các mối liên kết
- ⇒ **Bước 4:** Nhận dạng thuộc tính xác định cho mỗi kiểu thực thể
- ⇒ **Bước 5:** Nhận dạng các cấu trúc siêu kiểu/ kiểu con
- ⇒ **Bước 6:** Vẽ sơ đồ ER

Sơ đồ thực thể - liên kết

- ➔ Mô hình ER được diễn tả bằng sơ đồ liên kết thực thể (entity relationship diagram - ERD)
- ➔ Ba phần tử cơ bản:
 - Kiểu thực thể (entity Type)
 - Quan hệ (Relationship)
 - Các thuộc tính (Attribute)

Thực thể - Entity

- ⇒ Một *thực thể* (an entity) là một “sự vật” hoặc “đối tượng” mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Ví dụ như một nhân viên trong một tổ chức là một thực thể.
- ⇒ Thực thể có thể là
 - Một người như nhân viên, sinh viên,...
 - Một nơi chốn như thành phố, đất nước,...
 - Một sự kiện như mua hàng, trả lương,...
 - Một khái niệm như môn học, tài khoản,...

Kiểu thực thể - Entity Type

- ⇒ Một *kiểu thực thể hay tập thực thể (an entity set)* là một tập hợp các thực thể có cùng những tính chất hoặc thuộc tính.
 - Ví dụ như tập hợp tất cả những người mà họ là nhân viên của một tổ chức là một tập thực thể *nhanvien*.
- ⇒ Mỗi tập thực thể được đặt một tên gọi, thông thường là danh từ số ít và viết chữ in hoa.
 - Ví dụ như KHACHHANG, HOADON,...
- ⇒ Biểu diễn: bằng hình chữ nhật.

KHACHHANG

SANPHAM

Kiểu thực thể - Entity Type

- ⇒ **Thể hiện (instance)** của một kiểu thực thể là một trường hợp cụ thể của kiểu thực thể đó.
- ⇒ Ví dụ: kiểu thực thể Khách Hàng có các điển hình là Lan và Minh. Mỗi Khách Hàng đều có mã khách khác nhau, và có thể thực hiện các dịch vụ như đặt hàng, thanh toán tiền



Cách đặt tên và ký hiệu

- ⇒ Mỗi kiểu thực thể phải có một tên gọi, nên là danh từ số ít và viết chữ hoa.
- ⇒ Ký hiệu của các kiểu thực thể



EMPLOYEE

Thực thể mạnh



DEPENDENT

Thực thể yếu

Các kiểu thực thể

- ➔ **Kiểu thực thể mạnh** (strong entity type): tồn tại độc lập với những kiểu thực thể khác và có khóa chính
- ➔ **Kiểu thực thể yếu** (weak entity type): tồn tại phụ thuộc vào kiểu thực thể khác, có thể không có đủ các thuộc tính để cấu thành một khóa chính

EMPLOYEE

Thực thể mạnh

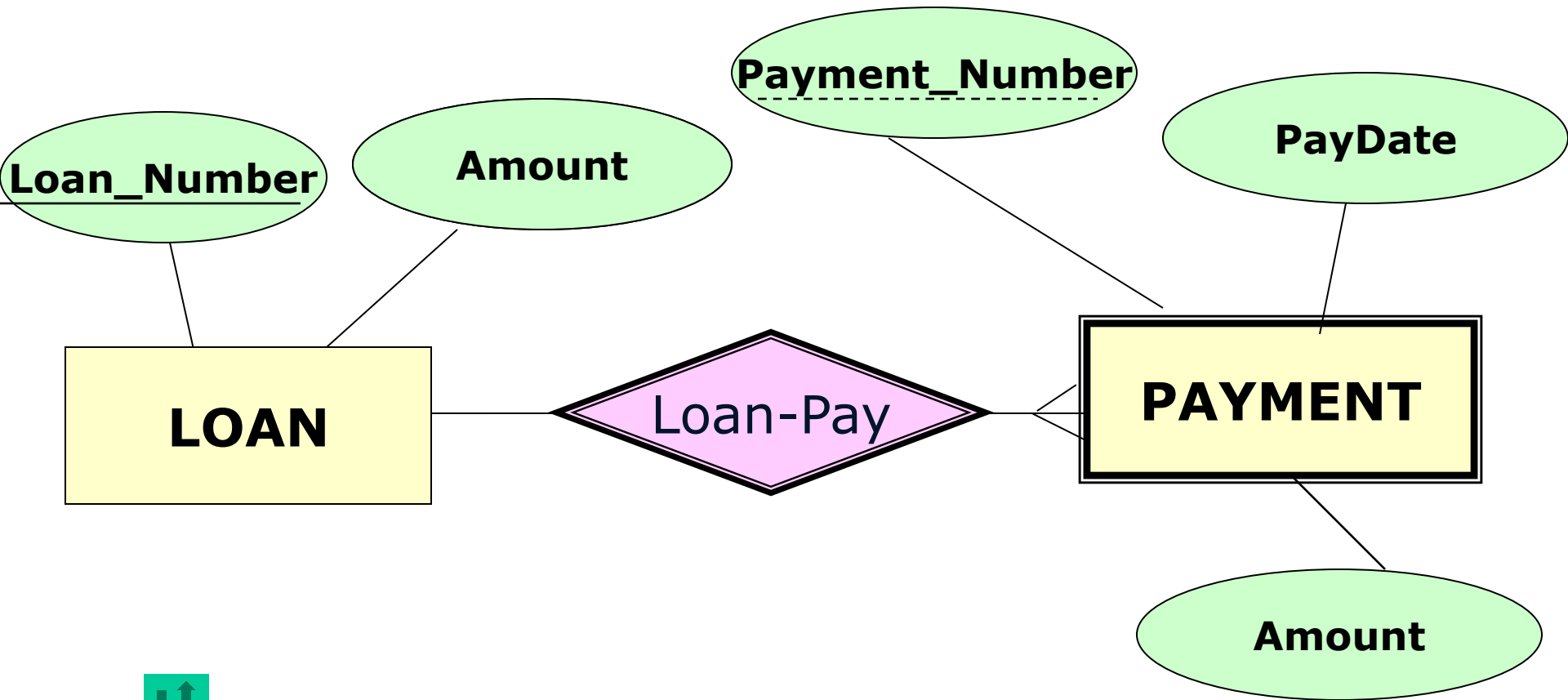
DEPENDENT

Thực thể yếu

➔ Ví dụ:

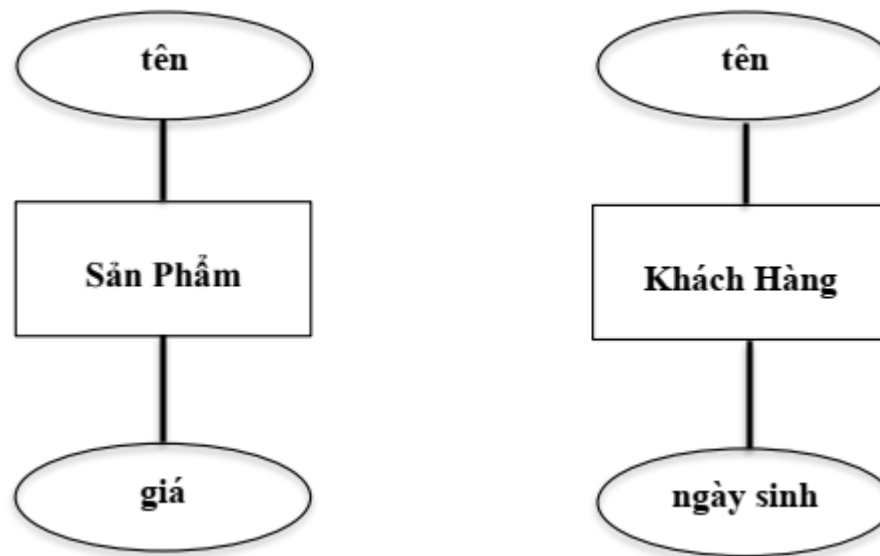
- LOAN (Mượn) là kiểu thực thể mạnh.
- PAYMENT (Trả) là kiểu thực thể yếu, lệ thuộc vào LOAN.
- Xác định thực thể mạnh và yếu cho 2 thực thể BenhNhan va ThanNhan

Ví dụ thực thể mạnh/yếu



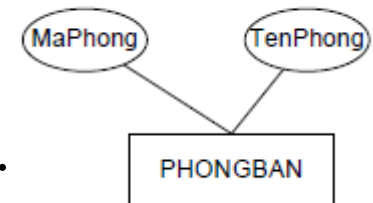
Thuộc tính - attribute

Một *thuộc tính* là một tính chất của thực thể, một đặc tính xác định phân biệt mà tất cả các thể hiện của thực thể có thể có. Ví dụ, thuộc tính Tên của thực thể **Khách Hàng**, thuộc tính Giá của thực thể **Sản Phẩm**



Thuộc tính - attribute

- ➔ Mỗi tập thực thể có 1 số thuộc tính.
- ➔ Thuộc tính là các đặc trưng (*properties*) được sử dụng để biểu diễn thực thể hay 1 mối liên kết.
- ➔ Được biểu diễn bằng *hình OVAL*.
- ➔ Các loại thuộc tính
 - Thuộc tính bắt buộc và thuộc tính tùy chọn.
 - Thuộc tính đơn và thuộc tính phức hợp.
 - Thuộc tính đơn trị và thuộc tính đa trị.
 - Thuộc tính chứa và thuộc tính dẫn xuất.
 - Thuộc tính khóa và thuộc tính không khóa.
 - Thuộc tính rỗng



Các kiểu thuộc tính

➔ **Thuộc tính đơn (simple attribute):** là thuộc tính không thể phân nhỏ được.

Ví dụ: Color, Weight, HorsePower, Masv, CMND, Phai

➔ **Thuộc tính phức hợp (composite attribute):** là thuộc tính được kết hợp của một số thành phần hay có thể chia thành các thuộc tính khác

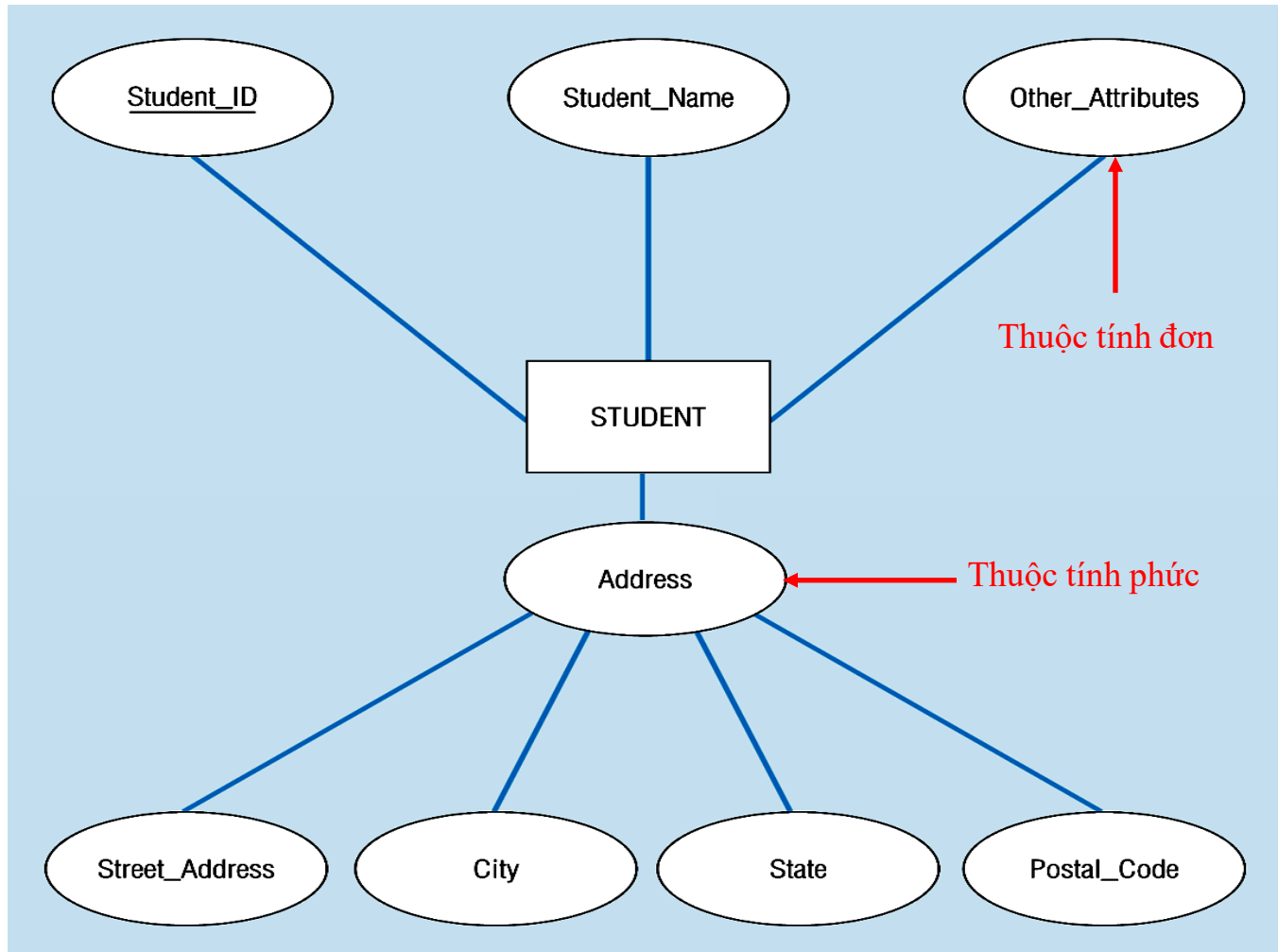
Ví dụ: Thuộc tính Address bao gồm các thành phần Street, District, City

▪ **Thuộc tính chứa (stored attribute):** là thuộc tính mà giá trị của nó không được suy dẫn từ các thuộc tính khác.

➔ **Thuộc tính dẫn xuất (derived attribute):** là thuộc tính mà trị của nó có thể tính ra được từ các thuộc tính khác

Ví dụ: Year_Employed là thuộc tính dẫn xuất từ thuộc tính Date_Employed

Các kiểu thuộc tính

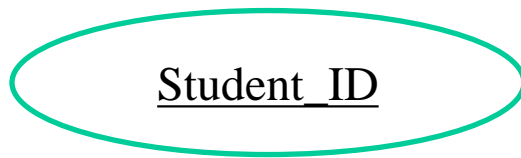


Các kiểu thuộc tính (tt)

- ➔ **Thuộc tính đơn trị (single valued attribute):** có 1 giá trị duy nhất tại một thời điểm.
- ➔ **Thuộc tính đa trị (multivalued attribute):** có nhiều giá trị tại một thời điểm.

Ví dụ: Thực thể COURSE có thuộc tính Teacher đa trị, một môn học có thể được dạy bởi nhiều hơn 1 thầy cô.

- ➔ **Thuộc tính rỗng.** Một giá rỗng (null value) được dùng đến khi một thực thể không có giá trị đối với một thuộc tính. Ví dụ một nhân viên nào đó không có người trong gia đình thì giá trị của thuộc tính *tên người trong gia đình* đối với nhân viên đó phải là rỗng.

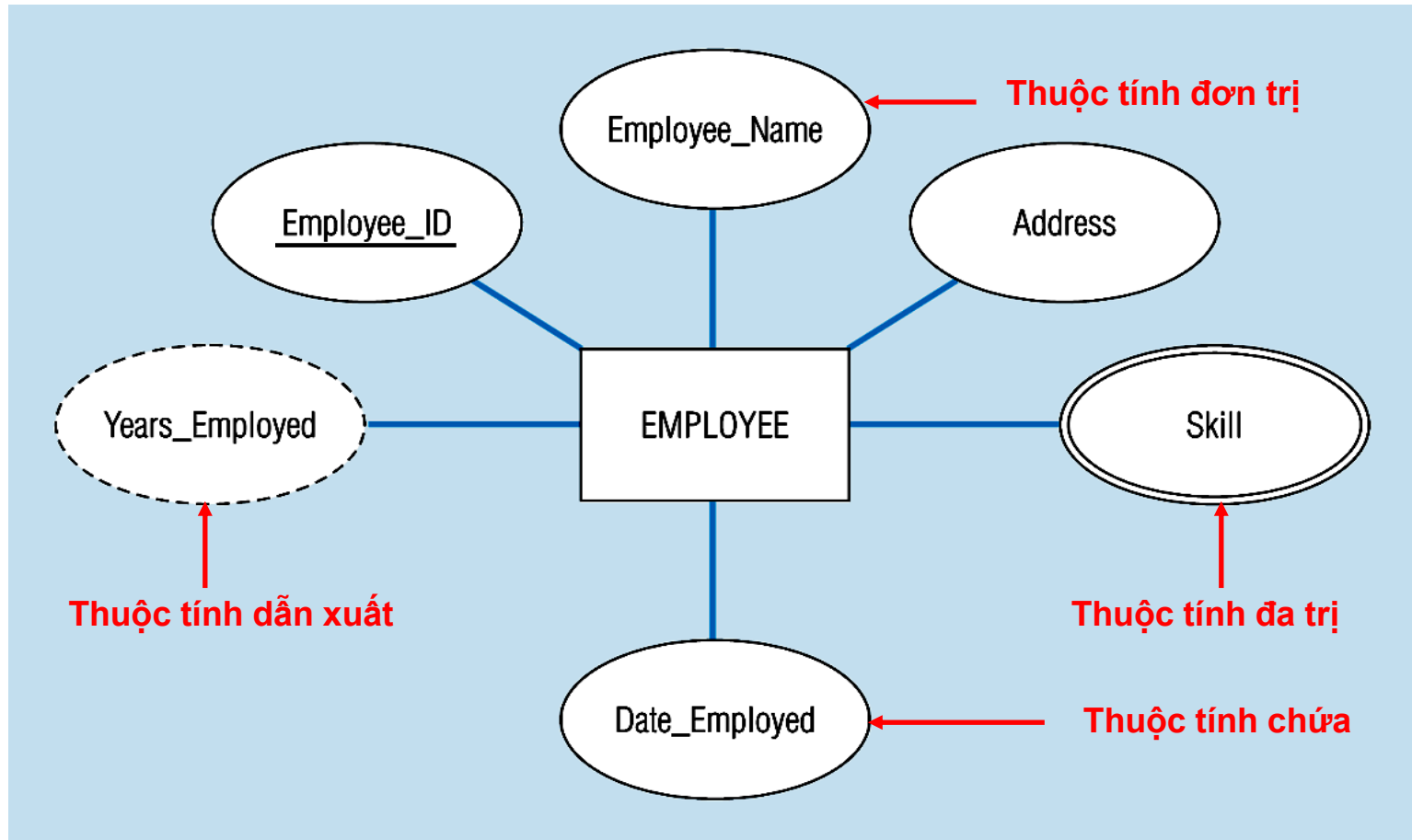


Thuộc tính xác định



Thuộc tính đa trị

Các kiểu thuộc tính (tt)



Các kiểu thuộc tính

⇒ Khóa / thuộc tính xác định (*key / identifier*)

- *Khóa* là một thuộc tính hoặc tổ hợp các thuộc tính dùng để xác định duy nhất một thể hiện của một kiểu thực thể.

⇒ Thuộc tính khóa và thuộc tính không khóa

- *Thuộc tính khóa (key attribute)* là thuộc tính ở trong khóa.
 - Thuộc tính khóa được gạch dưới.
- *Thuộc tính không khóa (non-key attribute)* là thuộc tính không ở trong khóa.
 - Thuộc tính không khóa còn được gọi là *thuộc tính mô tả (descriptor)*.



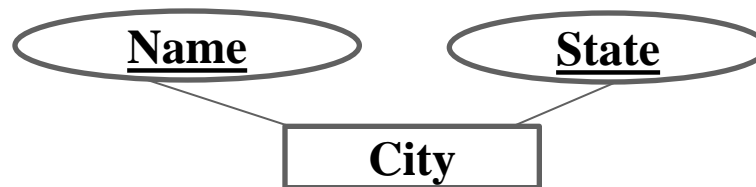
Các kiểu thuộc tính

⇒ Khóa đơn và khóa phức hợp

- *Khóa đơn (simple key)* là khóa chỉ có một thuộc tính.
- *Khóa phức hợp (composite key)* là khóa có nhiều hơn một thuộc tính.

⇒ Khóa dự tuyển (*candidate key*)

- *Khóa dự tuyển* là khóa của một tập thực thể.
- Một tập thực thể có ít nhất một khóa dự tuyển.

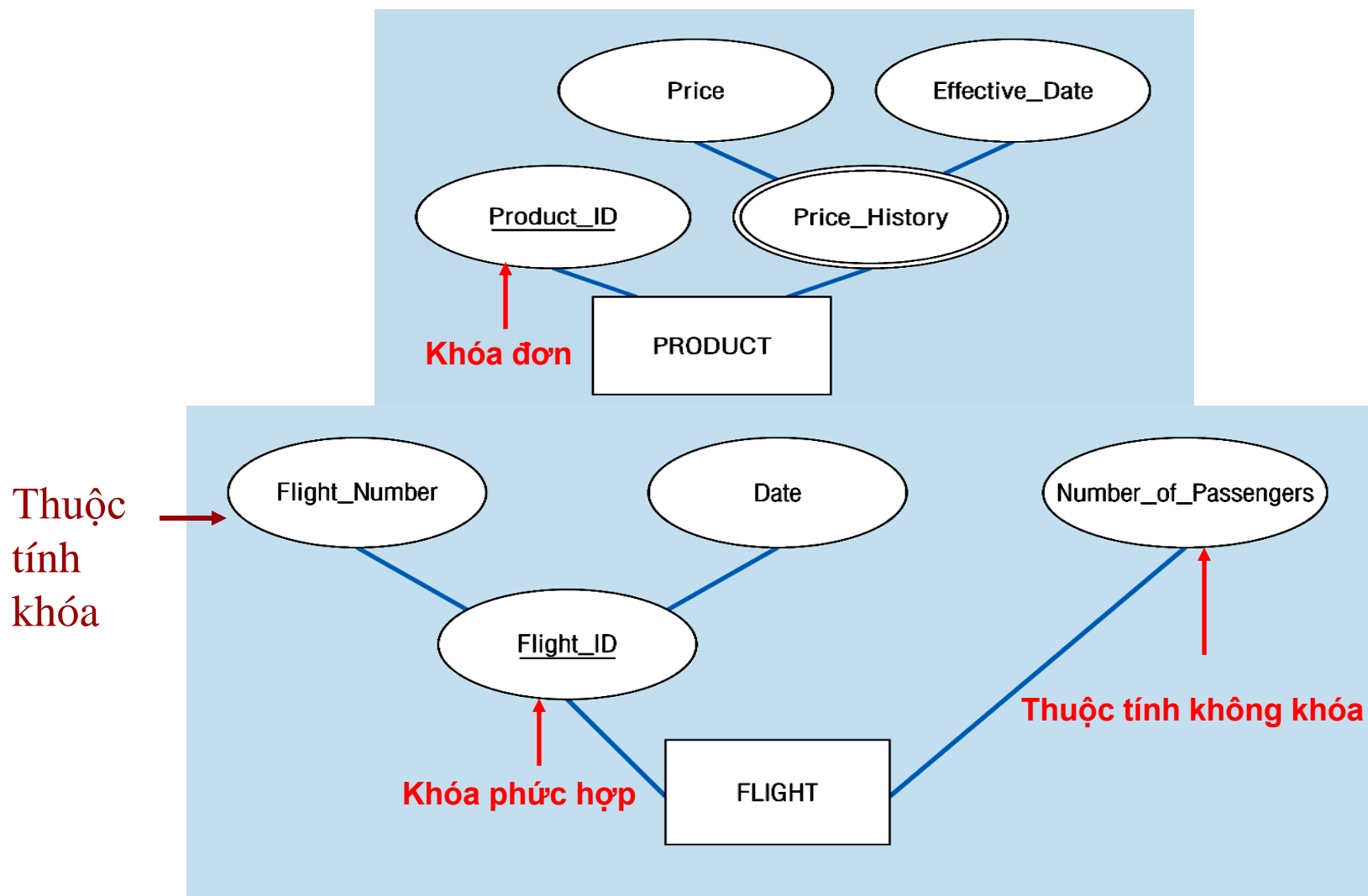


Các kiểu thuộc tính

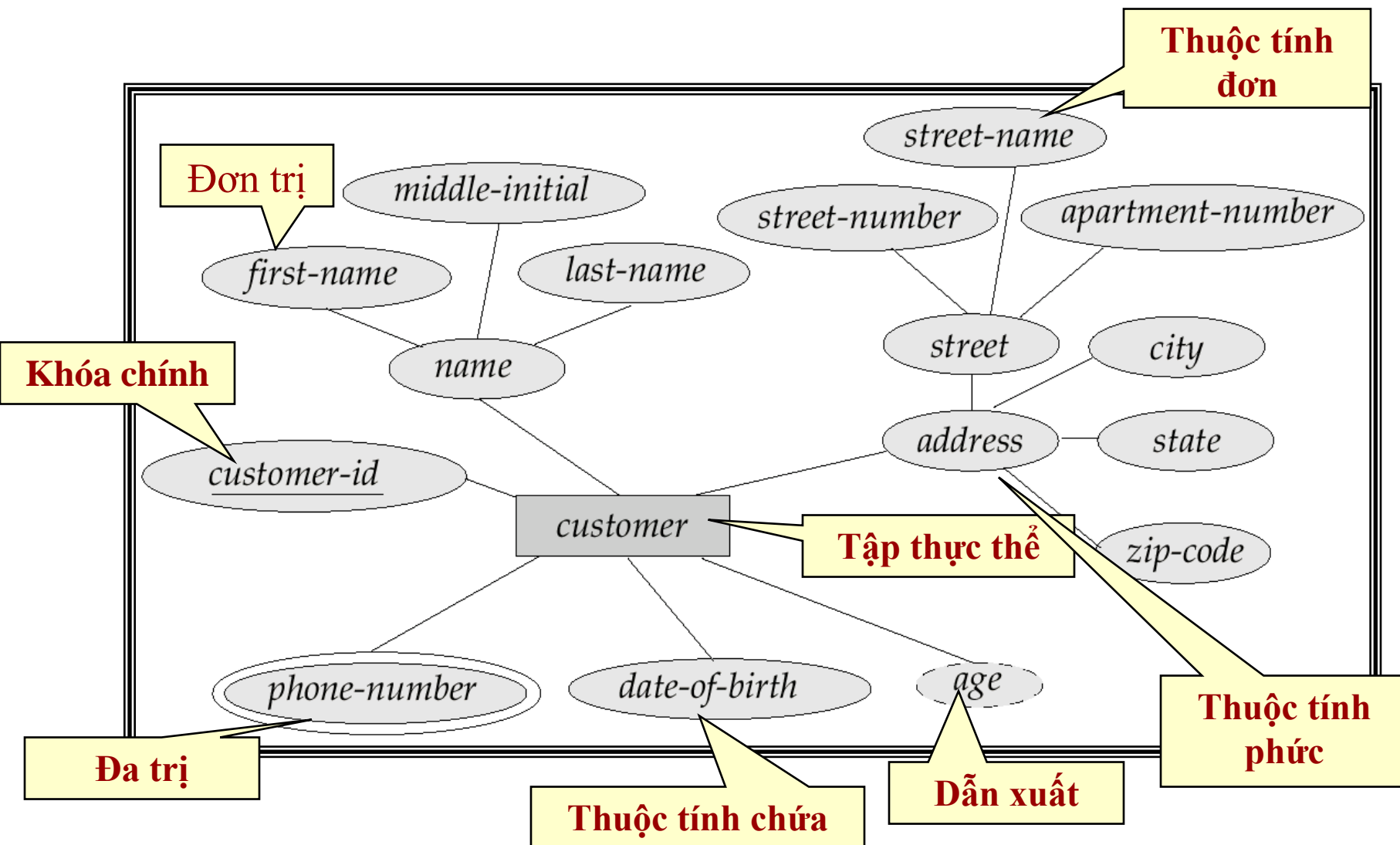
⇒ Khóa chính (*primary key*)

- *Khóa chính* là một khóa tiêu biểu trong các khóa dự tuyển của một kiểu thực thể.
- Một kiểu thực thể chỉ có một khóa chính.
- Khóa chính dùng để liên kết giữa các thực thể.

Các kiểu thuộc tính



Các kiểu thuộc tính



Bài tập

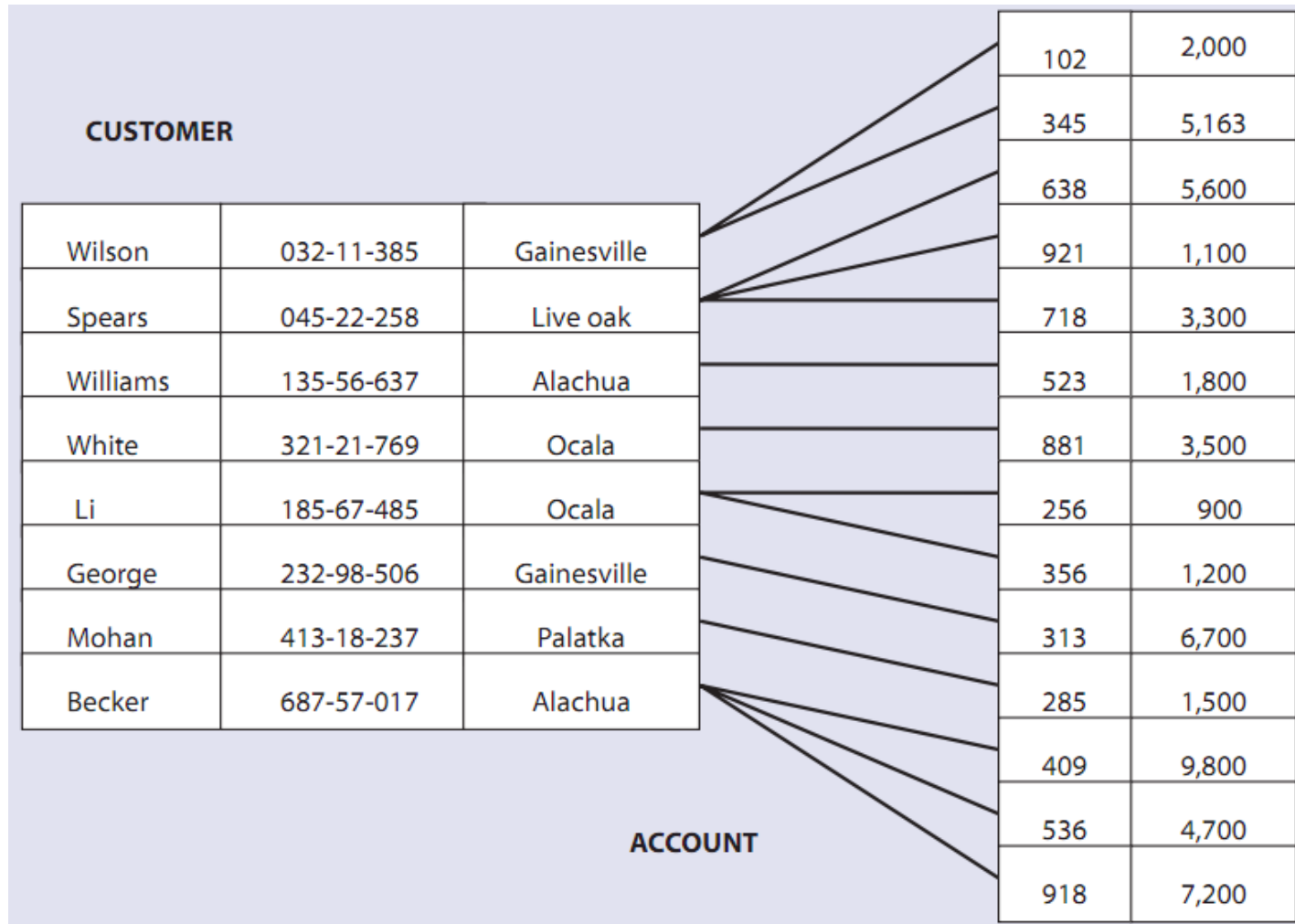
Hệ thống quản lý nhân viên

- Để quản lý việc phân công các nhân viên tham gia vào xây dựng các công trình. Công ty xây dựng X tổ chức quản lý như sau:
- Cùng lúc công ty có thể tham gia xây dựng nhiều công trình, mỗi công trình có một mã số công trình duy nhất (MACT), mỗi mã số công trình xác định các thông tin như: Tên gọi công trình (TENCT), địa điểm(ĐIADIEM), ngày công trình được cấp giấy phép xây dựng (NGAYCAPGP), ngày khởi công (NGAYKC), ngày hoàn thành (NGAYHT)
- Mỗi nhân viên của công ty ABC có một mã số nhân viên(MANV) duy nhất, một mã số nhân viên xác định các thông tin như: Họ tên (HOTEN), ngày sinh(NGAYSINH), phái (PHAI), địa chỉ (DIACHI), phòng ban, ...
- Công ty phân công các nhân viên tham gia vào các công trình, mỗi công trình có thể được phân cho nhiều nhân viên và mỗi nhân viên cùng lúc cũng có thể tham gia vào nhiều công trình. Với mỗi công trình một nhân viên có một số lượng ngày công (SLNGAYCONG) đã tham gia vào công trình đó.
- Công ty có nhiều phòng ban(Phòng kế toán, phòng kinh doanh, phòng kỹ thuật, phòng tổ chức, phòng chuyên môn, Phòng phục vụ,...). Mỗi phòng ban có một mã số phòng ban(MAPB) duy nhất, một phòng ban ứng với một tên phòng ban(TENPB)
- Xác định thực thể và thuộc tính mỗi thực thể

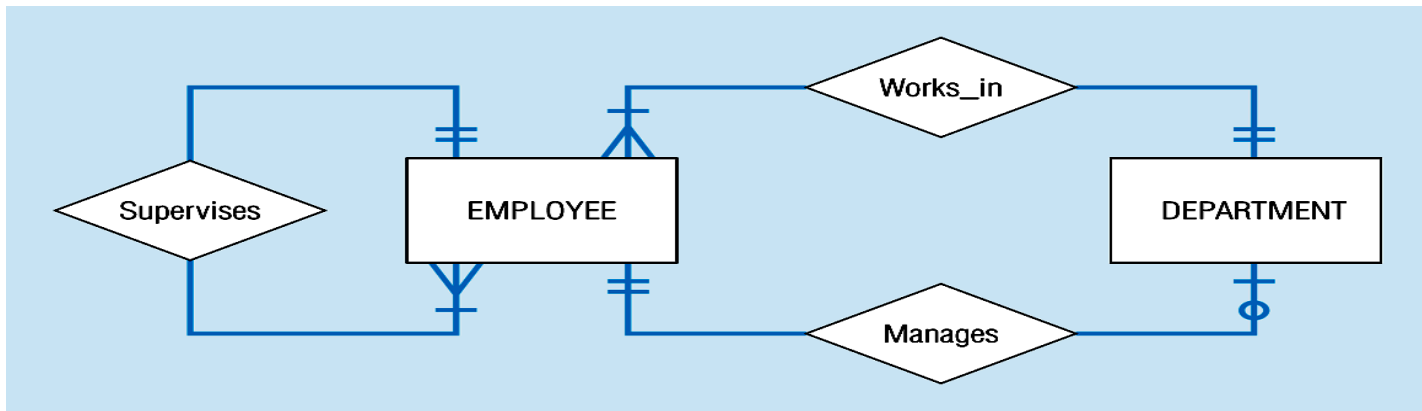
Mối liên kết - Relationship

- ➔ **Mối liên kết (relationship)** diễn tả sự kết hợp giữa một hay nhiều kiểu thực thể với nhau, là sự kết hợp biểu diễn sự tương tác giữa các điển hình (instance) của một hay nhiều kiểu thực thể (entity type)
- ➔ **Kiểu liên kết (relationship type)** là một sự kết hợp có ý nghĩa giữa các kiểu thực thể. Được biểu diễn bằng hình thoi.
- ➔ **Một điển hình liên kết (relationship instance)** là một sự kết hợp giữa các điển hình thực thể nơi mà mỗi điển hình liên kết bao gồm chính xác 1 thực thể từ mỗi kiểu thực thể tham gia vào.

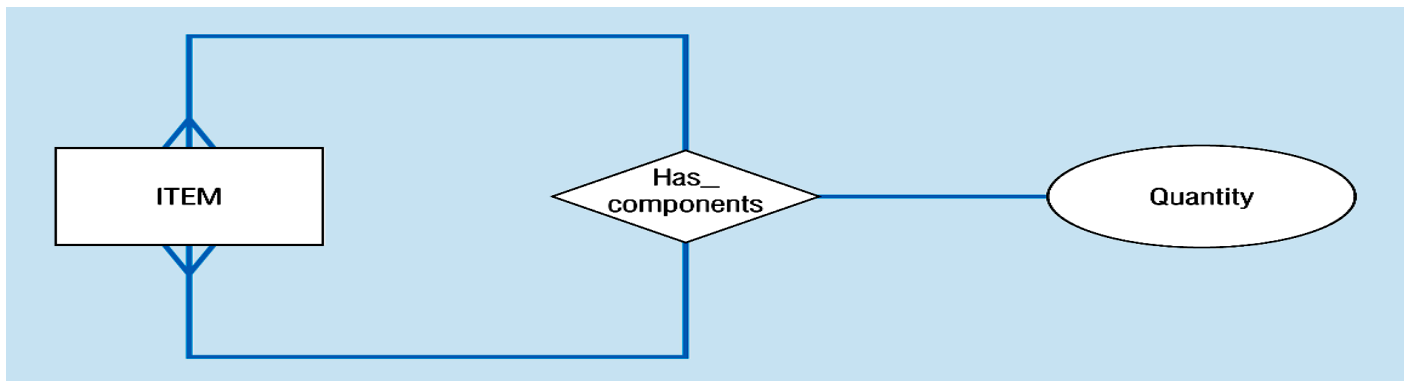
Mối liên kết - Relationship



Mối liên kết - Relationship

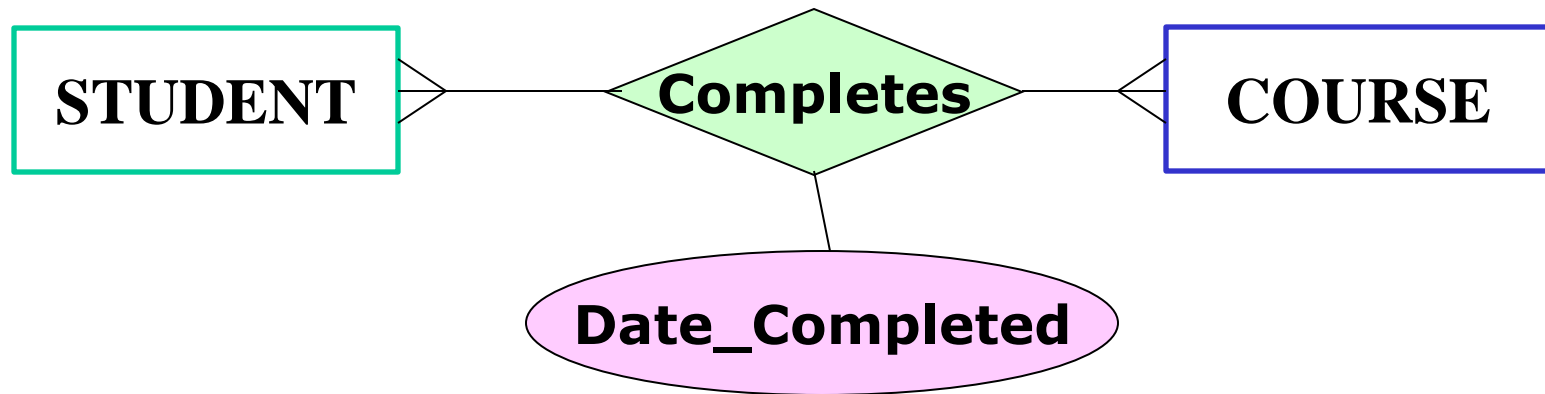


Hai thực thể có nhiều mối liên kết



Mối liên kết có thuộc tính

Ví dụ



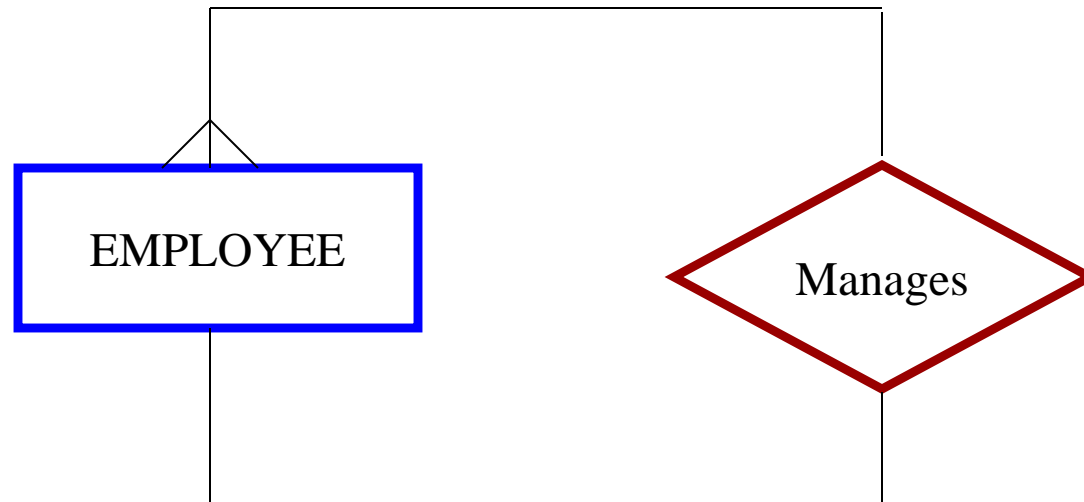
- Thuộc tính **Date_Completed** nên đặt ở đâu trong lược đồ trên?
 - Là 1 thuộc tính của mỗi liên kết **Completed** (thích hợp hơn là thuộc tính của 2 thực thể **STUDENT** và **COURSE**)

Bậc và các kiểu liên kết

- ⇒ Bậc của mỗi liên kết (*degree / arity of relationship*): là số lượng kiểu thực thể tham gia vào mỗi liên kết
- ⇒ Các kiểu liên kết
 - Liên kết 1 ngôi (*unary relationship*)
 - Liên kết 2 ngôi (*binary relationship*)
 - Liên kết 3 ngôi (*ternary relationship*): 3 kiểu thực thể đồng thời tham gia vào mỗi liên kết.

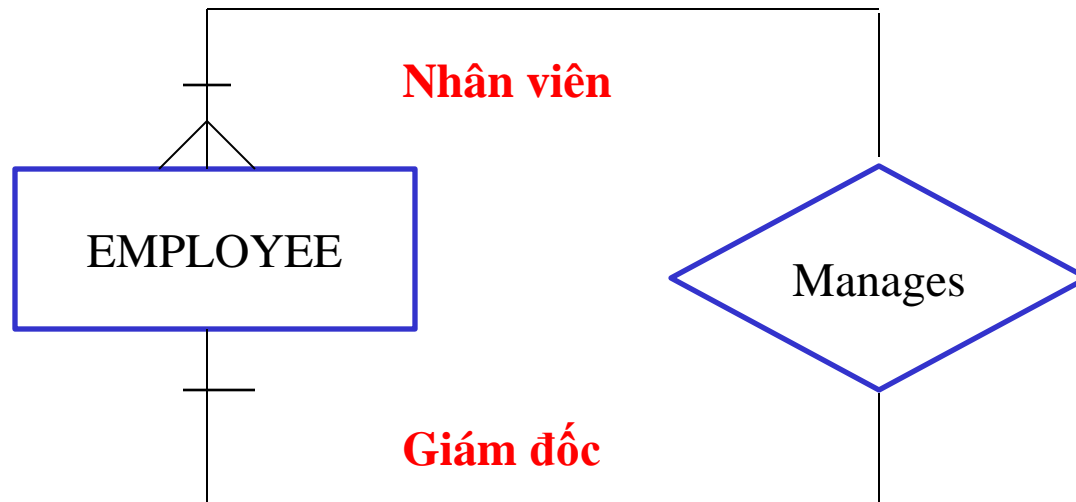
Liên kết một ngôi - Unary relationship

- ⇒ Là mối quan hệ giữa cùng 1 kiểu thực thể.
- ⇒ Còn gọi là mối liên kết đệ quy (recursive relationship)
- ⇒ Ví dụ:

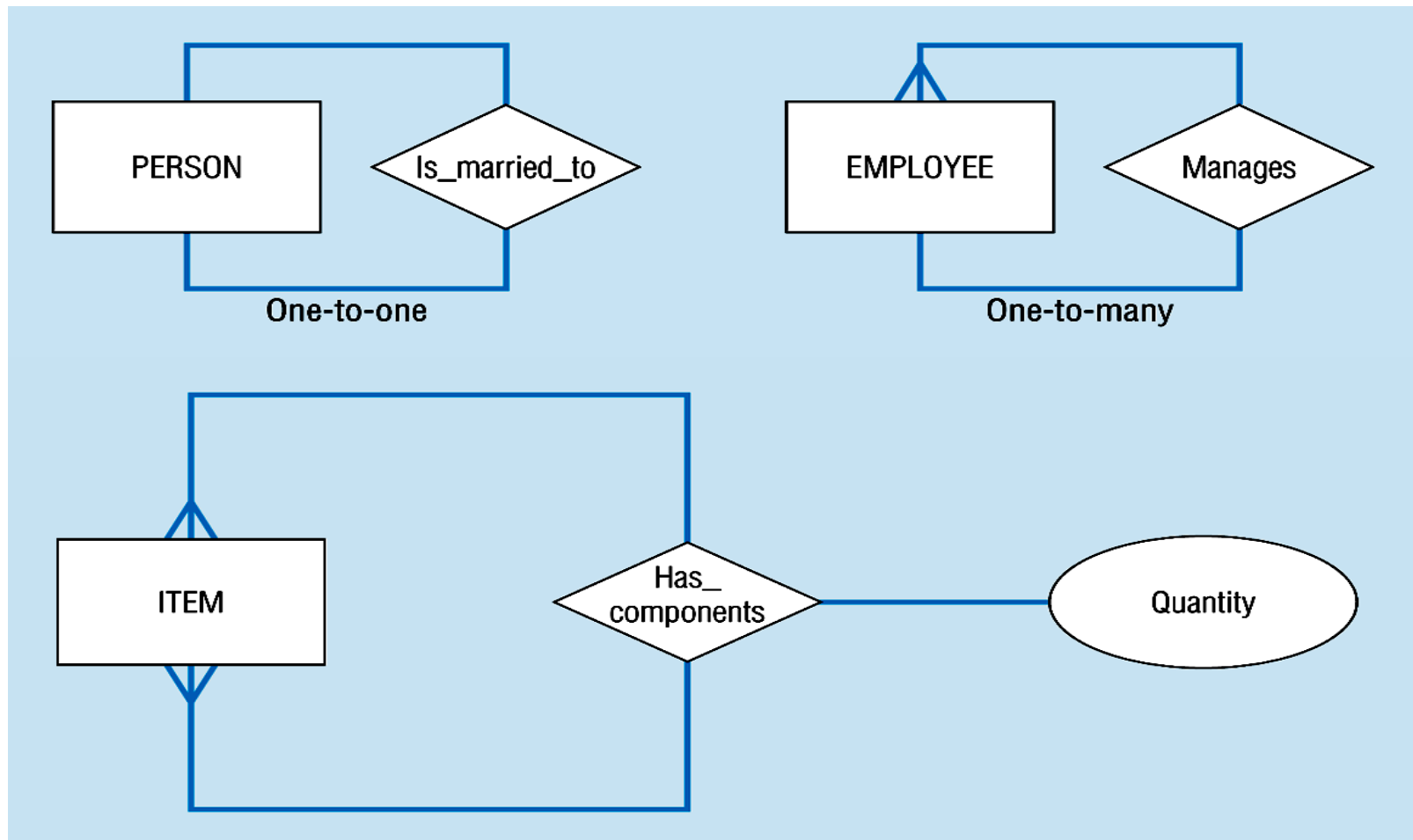


Liên kết một ngôi - Unary relationship

- ⇒ Đôi khi một thực thể xuất hiện nhiều hơn 1 lần trong mỗi quan hệ.
- ⇒ Để phân biệt, nên tạo role (nhãn) trên các cạnh nối giữa mỗi quan hệ và thực thể.



Liên kết một ngôi - Unary relationship



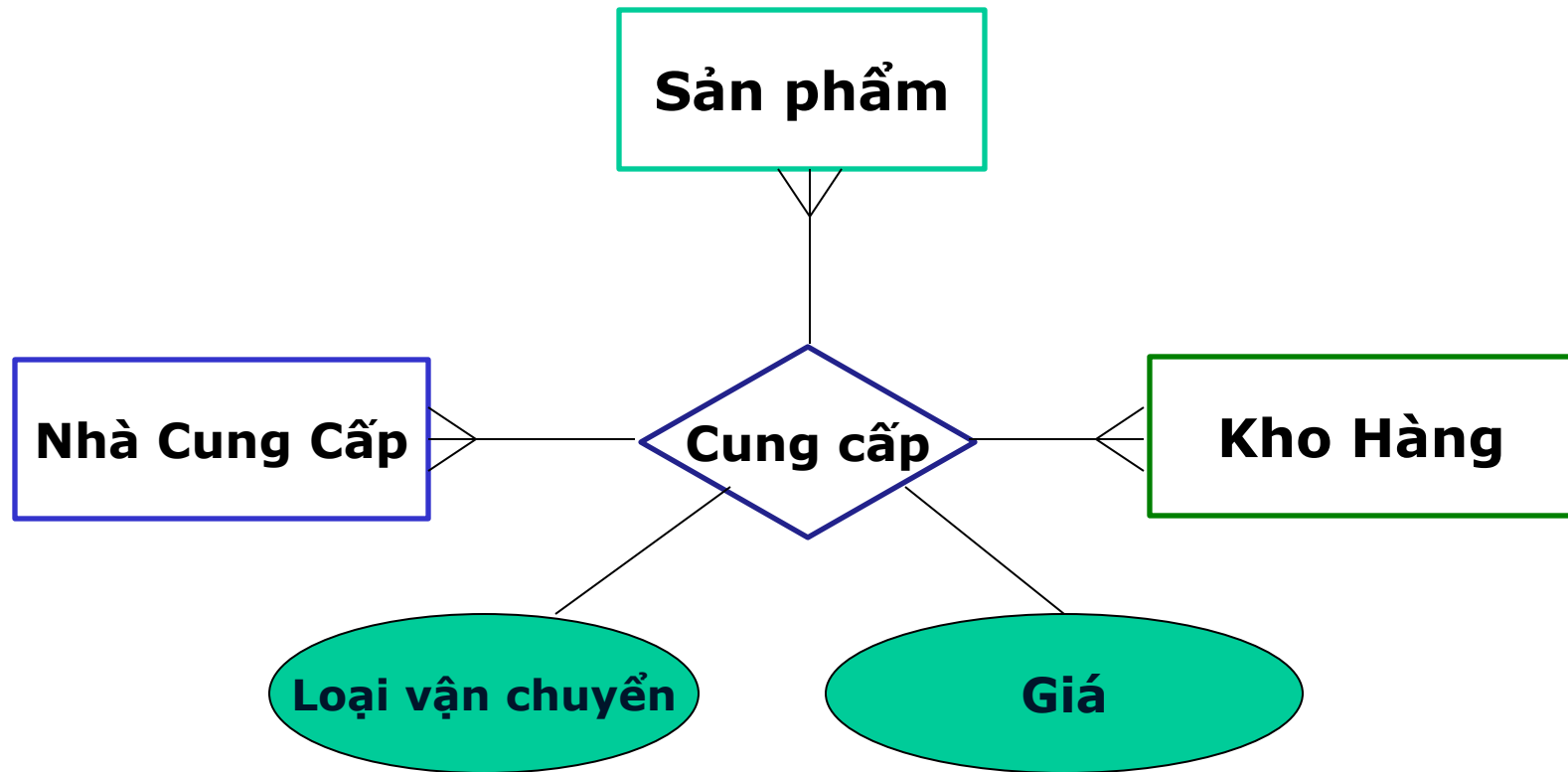
Liên kết hai ngôi - Binary relationship

⇒ Là mối liên kết giữa hai kiểu thực thể



Liên kết ba ngôi - Ternary relationship

⇒ Là mối liên kết giữa 3 kiểu thực thể



Lượng số của mối liên kết - Cardinality

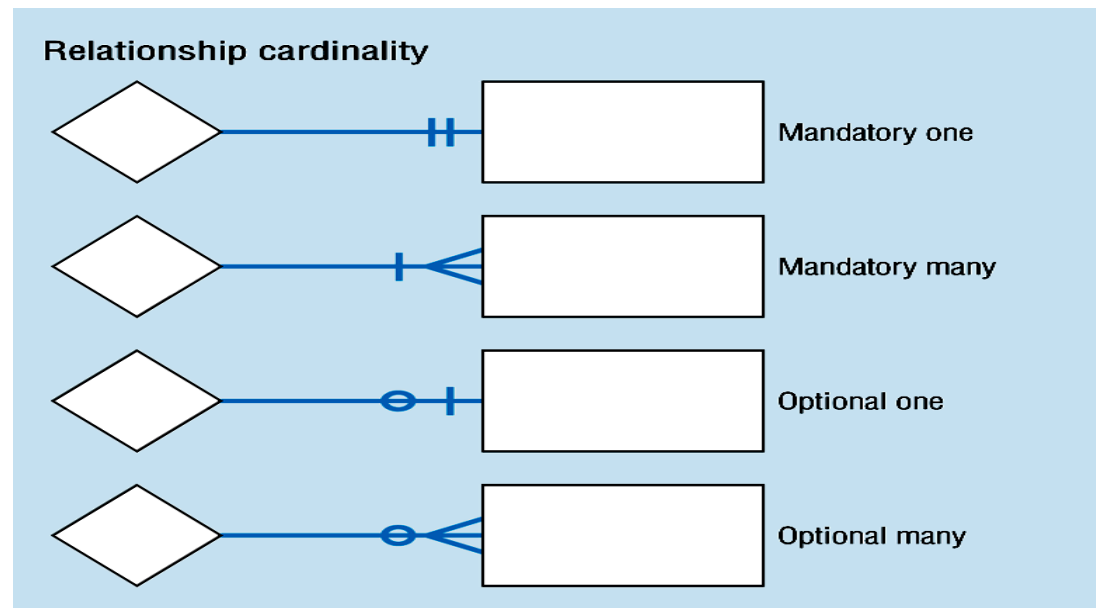
- ⇒ Lượng số là số thể hiện của kiểu thực thể B mà có thể liên kết với mỗi thể hiện của kiểu thực thể A
- ⇒ Lượng số tối thiểu (minimum cardinality): bằng 0 hoặc 1, là số lần tối thiểu mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.
- ⇒ Lượng số tối đa (maximum cardinality): bằng 1 hoặc n, là số lần tối đa mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.

Lượng số của mỗi liên kết - Cardinality

- ➔ Nếu lượng số tối thiểu là 0, kiểu thực thể B được gọi là nhiệm ý. Ký hiệu là O
- ➔ Nếu lượng số tối thiểu và tối đa đều là 1 thì lượng số này được gọi là bắt buộc (mandatory). Ký hiệu là ||

➔ Ba dạng liên kết:

- Liên kết 1-1
- Liên kết 1-n
- Liên kết n-n

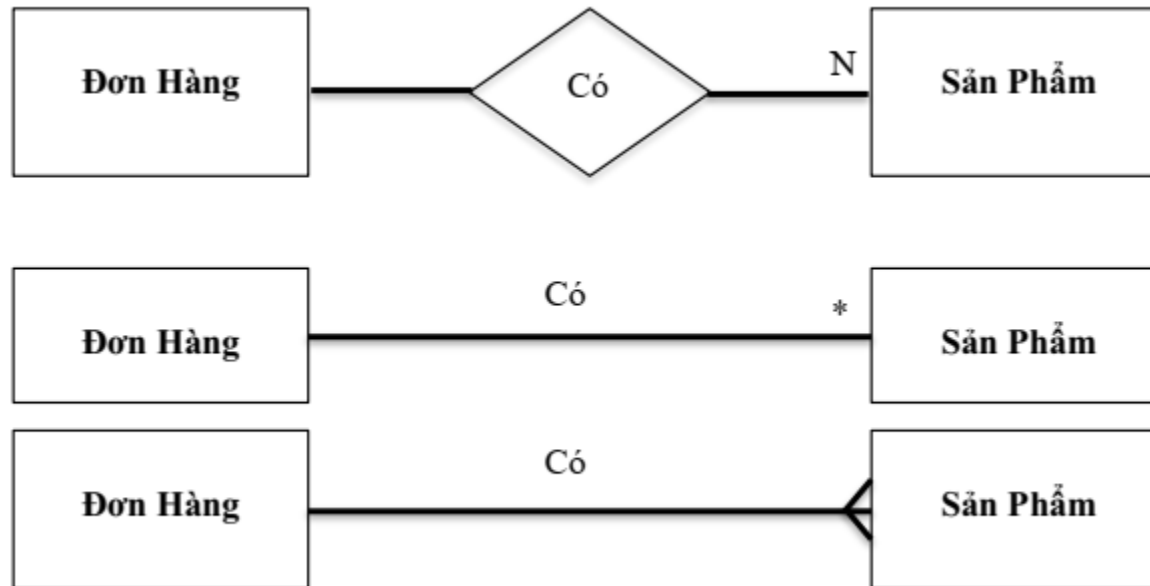


Lượng số của mối liên kết - Cardinality

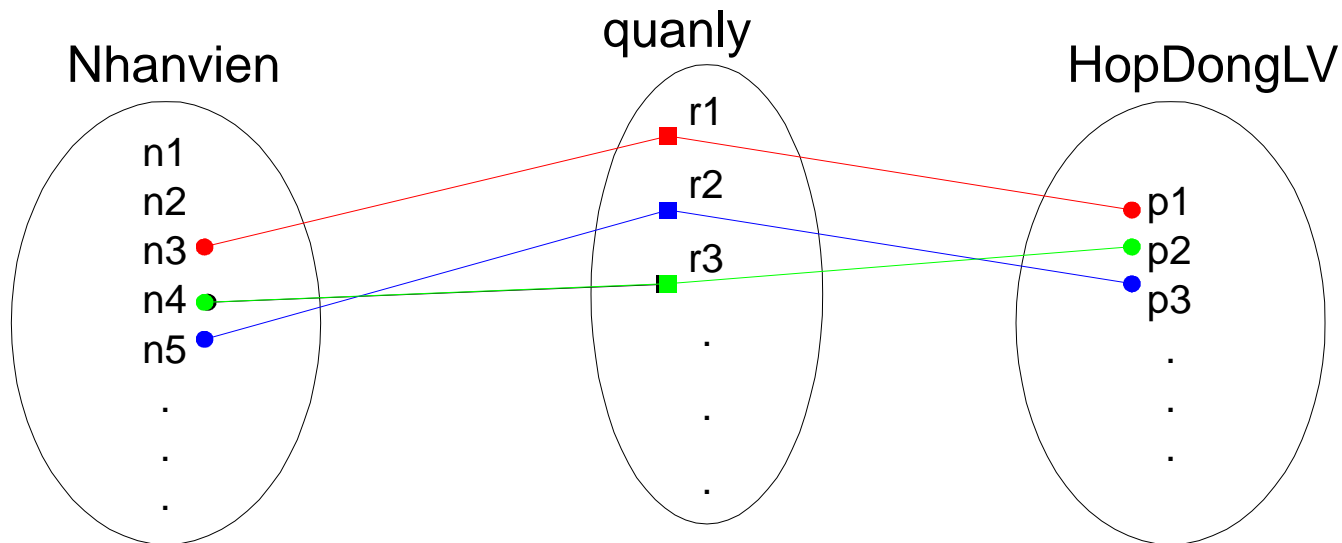
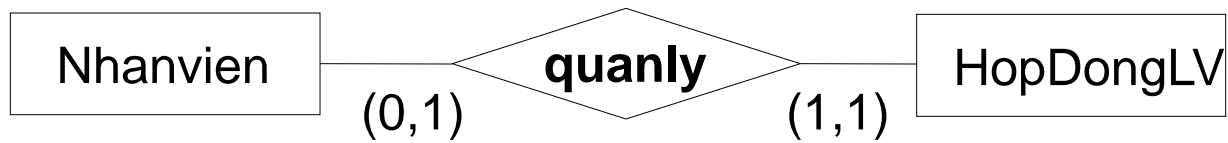
⇒ Các loại lượng số

- **một - một (*one-to-one*)**: một thể hiện của kiểu thực thể a liên kết với một thể hiện của kiểu thực thể b và ngược lại
- **một - nhiều (*one-to-many*)**: một thể hiện của kiểu thực thể a liên kết với nhiều thể hiện của kiểu thực thể b ; ngược lại một thể hiện của kiểu thực thể b chỉ liên kết với một thể hiện của kiểu thực thể a .
- **nhiều - nhiều (*many-to-many*)**: một thể hiện của kiểu thực thể a liên kết với nhiều thể hiện của kiểu thực thể b ; ngược lại một thể hiện của kiểu thực thể b liên kết với nhiều thể hiện của kiểu thực thể a .

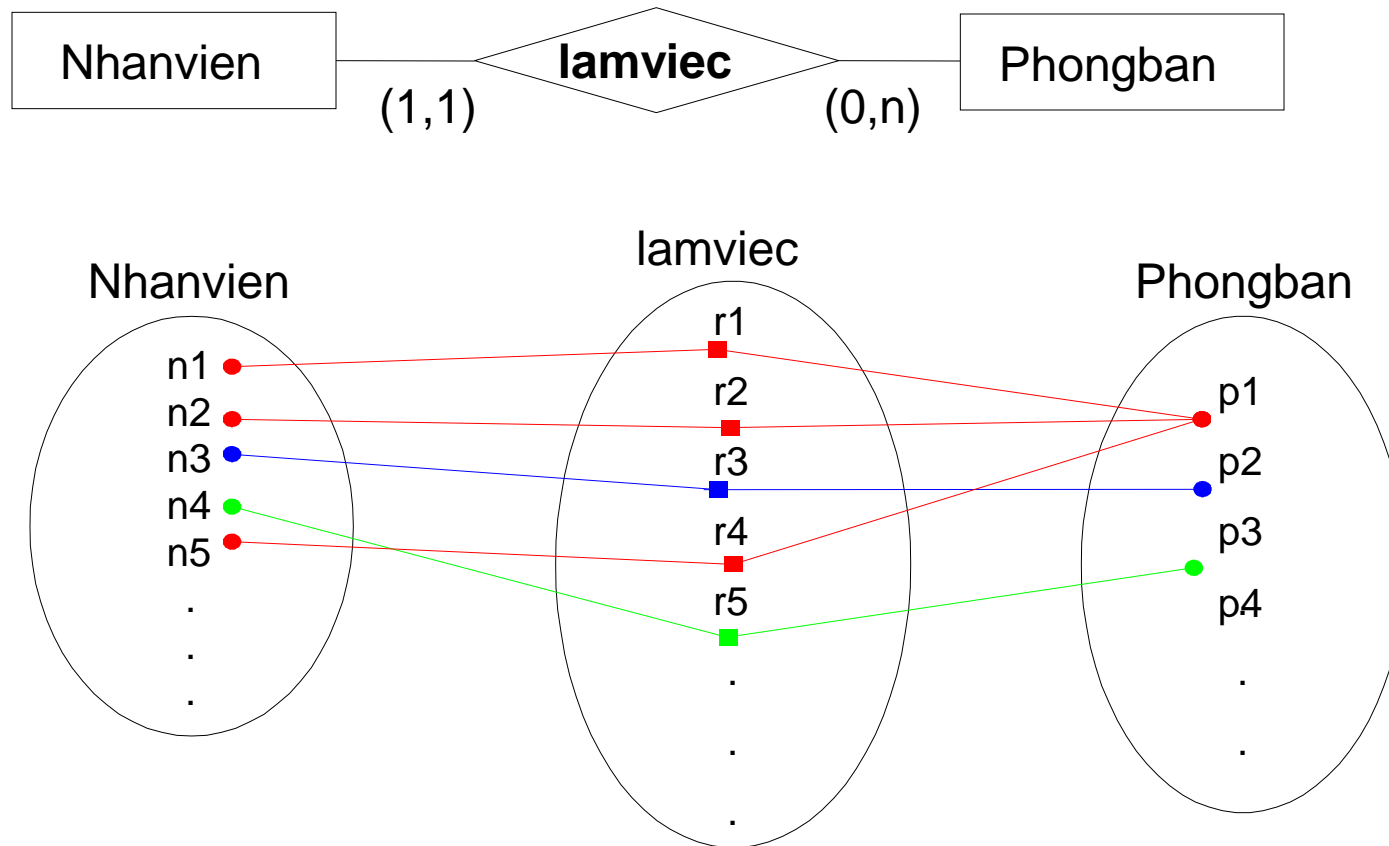
Quan hệ một-nhiều giữa các thực thể



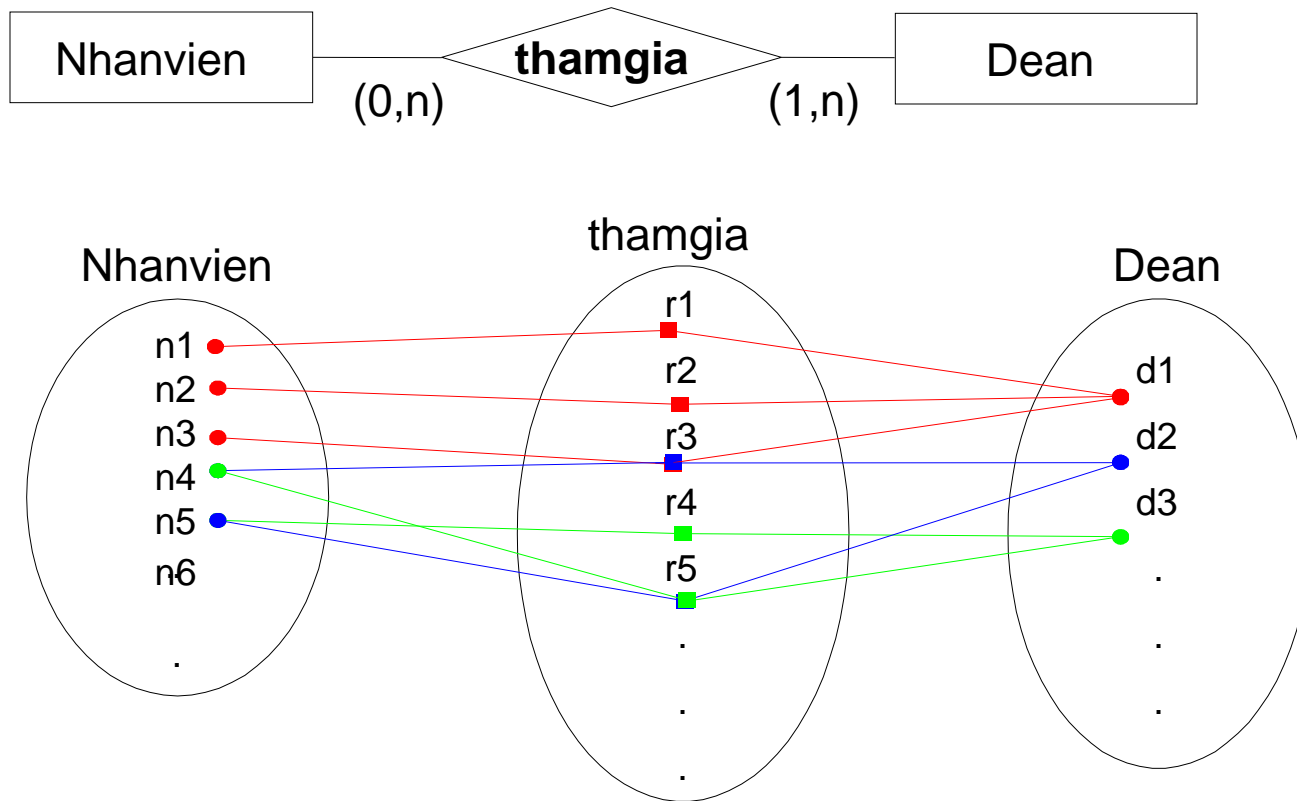
Mối Kết hợp 1-1



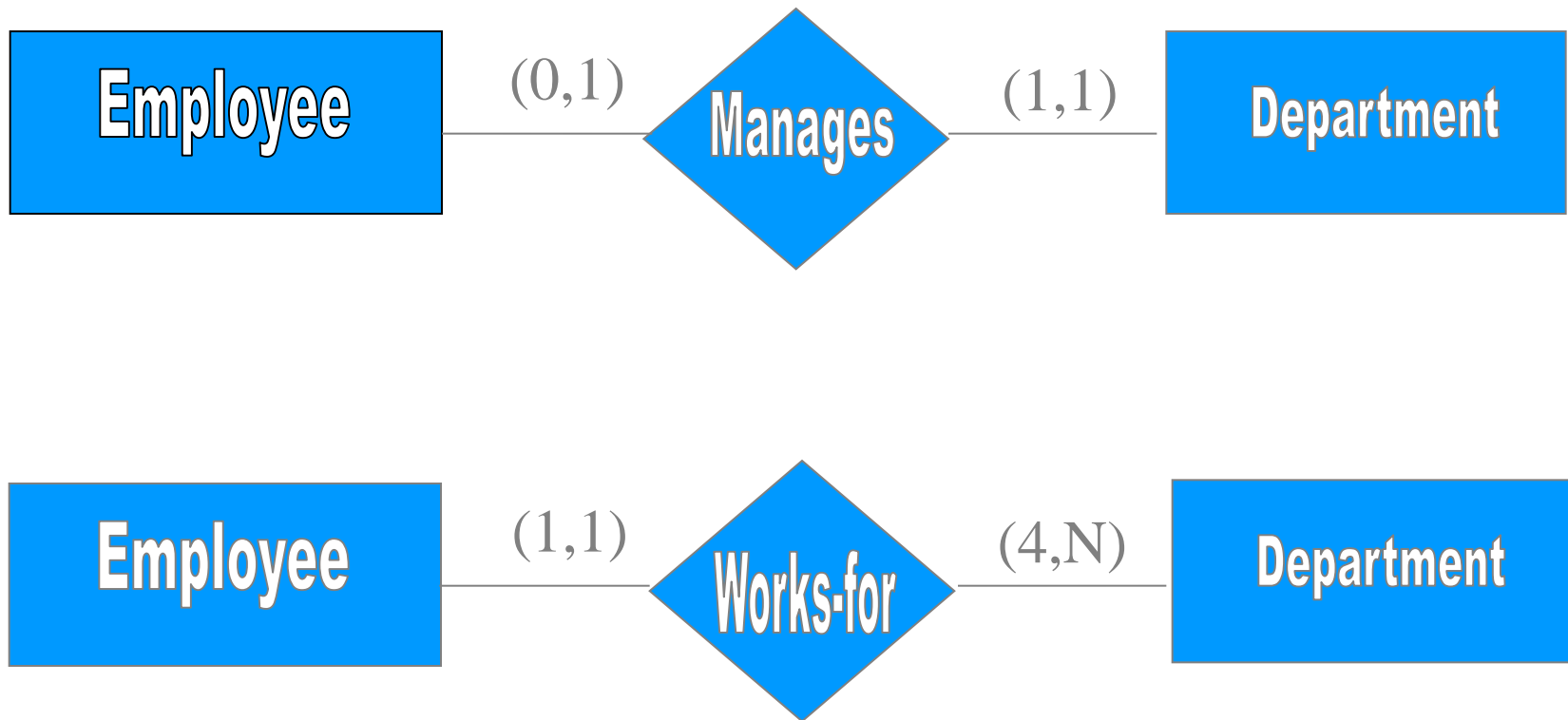
Mối Kết hợp 1-n



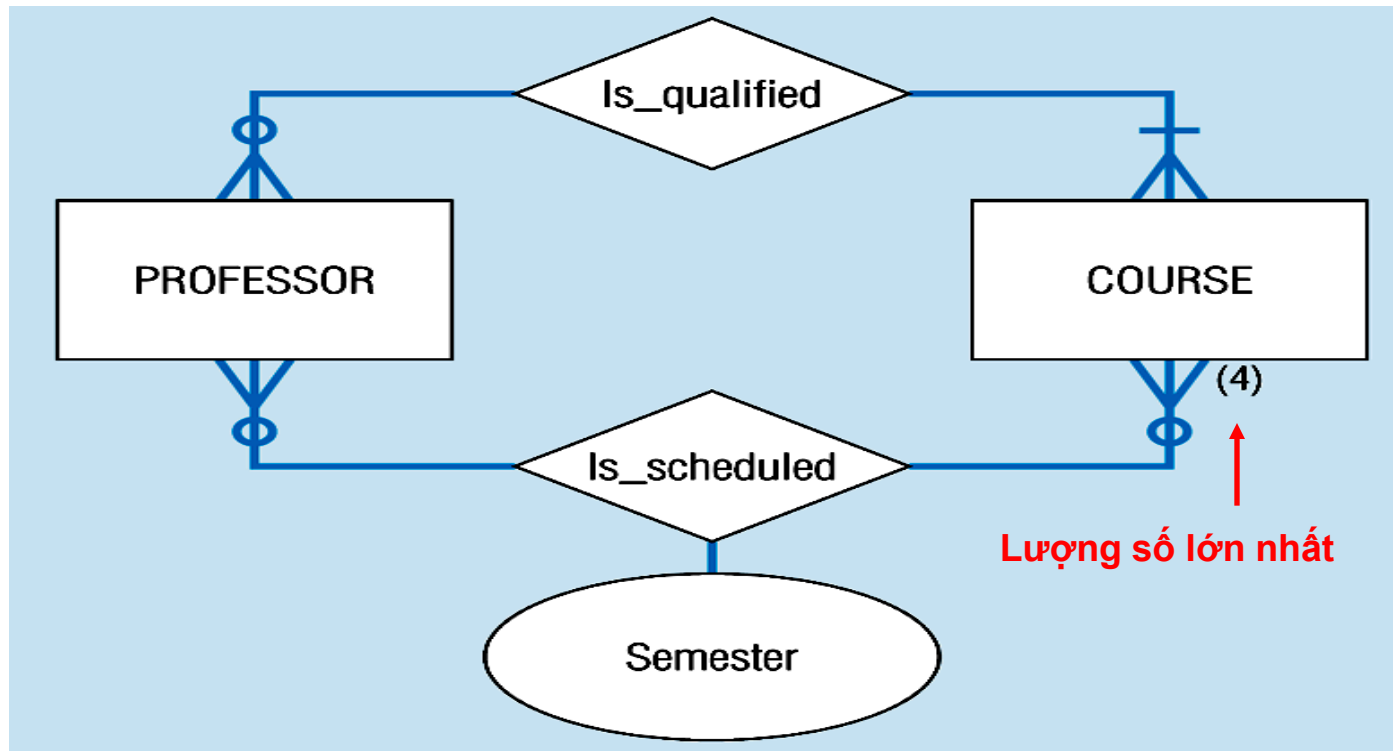
Mối Kết hợp n-n



Ví dụ mối liên kết



Ví dụ mối liên kết



Mối liên kết có lượng số tối đa xác định

XIN CẢM ƠN!